VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2017 Pavel Had

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

VYTVOŘENÍ INTERAKTIVNÍCH PROGRAMŮ PRO PODPORU VÝUKY ZPRACOVÁNÍ SIGNÁLŮ A OBRAZŮ
INTERACTIVE SOFTWARE TOOLS FOR TEACHING SIGNAL AND IMAGE PROCESSING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE Pavel Had
AUTHOR

VEDOUCÍ PRÁCE Ing. Marie Mangová
SUPERVISOR

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Pavel Had ***ID:*** 147433
Ročník: 3 ***Akademický rok:*** 2016/17

NÁZEV TÉMATU:

Vytvoření interaktivních programů pro podporu výuky zpracování signálů a obrazů

POKYNY PRO VYPRACOVÁNÍ:

Vytvořte programy pro interaktivní podporu výuky zejména v kurzech Analýza signálů a soustav, Číslicové zpracování signálů a Základy počítačové sazby a grafiky. Budou napsány v JavaScriptu a tématicky zaměřeny na: 1/ lineární kombinaci obrazů, 2/ metodu nejmenších čtverců a lineární regresi, 3/ diskrétní lineární konvoluci ve 2D, 4/ interpolaci v 1D.

DOPORUČENÁ LITERATURA:

[1] Žára, J.; Beneš, B.; Sochor, J.; Felkel, P.: Moderní počítačová grafika (2. vydání), Computer Press, 2005, ISBN 80-251-0454-0.

[2] Anděl, J.: Matematická statistika. 1. vyd. Praha: SNTL/ALFA, 1978.

Termín zadání: 1.2.2017 ***Termín odevzdání:*** 8.6.2017

Vedoucí práce: Ing. Marie Mangová
Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Tato práce se zabývá návrhem a programováním čtyř webových appletů pro pozdější výuku. Jsou zde použity programovací jazyky HTML, CSS a JavaScript.

KLÍČOVÁ SLOVA

Applet, HTML, CSS, JavaScript, Metoda nejmenších čtverců, Lineární regrese, Diskrétní lineární konvoluce, Interpolace

ABSTRACT

This thesis describes the design and programming of the four web applets for later learning. There are used programming languages HTML, CSS and JavaScript.

KEYWORDS

Applet, HTML, CSS, JavaScript, Least squares method, Linear regression, Discrete linear convolution, Interpolation

HAD, Pavel *Vytvoření interaktivních programů pro podporu výuky zpracování signálů a obrazů*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, Rok. 29 s. Vedoucí práce byl Ing. Marie Daňková

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Vytvoření interaktivních programů pro podporu výuky zpracování signálů a obrazů" jsem vypracoval(a) samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor(ka) uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil(a) autorská práva třetích osob, zejména jsem nezasáhl(a) nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom(a) následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu semestrální práce slečně Ing. Marii Daňkové za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno . . . .

podpis autora(-ky)

OBSAH

SEZNAM OBRÁZKŮ

SEZNAM VÝPISŮ

ÚVOD

Tato práce se zabývá tvorbou interaktivních appletů pro budoucí žáky školy VUT, především pro předmět BASS (Analýza signálů a soustav), BCZS (Číslicové zpracování signálů) a BZSG (Základy počítačové sazby a grafiky).

V práci jsou obsaženy tyto 4 výukové applety: 1/ lineární kombinace obrazů, 2/ metoda nejmenších čtverců a lineární regrese, 3/ diskrétní lineární konvoluce ve 2D, 4/ interpolace v 1D.

1 LINEÁRNÍ KOMBINACE OBRAZŮ

1.1 Lineární kombinace

Lineární kombinace je pojem, který označuje základní koncept studovaný oborem zvaný lineární algebra. Je to zobecněný pojem pro násobení a sčítání čísel. Pojem lineární kombinace slouží k definování dalších důležitých objektů v lineární algebře, jako je lineární oblast, lineární závislost apod.

1.2 Definice lineární kombinace

Uvažujme vektorový prostor V nad tělesem T. dále nechť $\vec{x} \in V$ je nějaký vektor a $\vec{x}_1, \ldots, \vec{x}_k$ je soubor k vektorů z prostoru V. Pak říkáme, že vektor $\vec{x}$ je lineární kombinací vektorů $\vec{x}_1, \ldots, \vec{x}_k$, právě když existuje k-tice čísel z tělesa $\alpha_1, \ldots, \alpha_k \in T$ taková, že lze vektor $\vec{x}$ vyjádřit ve tvaru sumy

$$\vec{x} = \alpha_1 \vec{x}_1 + \ldots + \alpha_k \vec{x}_k = \sum_{i=1}^{k} \alpha_i \vec{x}_i. \quad [4] \tag{1.1}$$

Číslům α_i ze vztahů výše říkáme koeficienty lineární kombinace. Jsou-li koeficienty α_i nulové, je lineární kombinace označována jako triviální. Takováto lineární kombinace je bez ohledu na hodnotu vektorů $\vec{x}_1, \ldots, \vec{x}_k$ vždy rovna nulovému vektoru. Je-li alespoň jeden z koeficientů $\alpha_i \neq 0$, pak říkáme, že lineární kombinace je netriviální.

1.3 Speciální případy lineární kombinace

Někdy je lineární kombinace příliš obecný pojem, tudíž se zavádějí konkretní kombinace, jako afinní a konvexní.

1.3.1 Afinní kombinace

Je to speciální druh lineární kombinace. Lineární kombinaci nazýváme afinní kombinace, právě když je součet jejích koeficientů roven jedné, neboli

$$\sum_{i=1}^{k} \alpha_i = 1. \quad [4] \tag{1.2}$$

1.3.2 Konvexní kombinace

Lineární kombinaci nazýváme konvexní kombinace, právě když je součet jejích koeficientů roven jedné a přitom jsou všechny koeficienty nezáporné, neboli

$$\sum_{i=1}^{k} \alpha_i = 1 \quad \cap \quad (\forall i \in 1, \ldots, k)(\alpha_i \geq 0) \quad [4] \tag{1.3}$$

1.4 Reprezentace obrazu v počítačové grafice

Obraz lze obecně reprezentovat několika grafickými modely. Jsou to modely RGB, CMY, CMYK, HSV, HSB, HSL, YUV. V této práci je použit pouze model RGB, ostatními se tudíž zabývat nebudeme.

1.4.1 Model RGB

RGB (Red, Green, Blue) je aditivní barevný model. Je založen na principu vnímání lidského oka, které je citlivé na tři základní barvy (červenou, zelenou a modrou).

Aditivní barevný model znamená, že se jednotlivé složky barev sčítají. Výsledné světlo má vyšší intenzitu než jednotlivé původní složky. Maximální součet všech tří složek dá barvu bílou.

1.4.2 Reprezentace barevných bodů v počítači

Jednotlivé barevné body, takzvané pixely jsou nejčastěji reprezentovány jako hodnoty jednotlivých barevných složek, červené, zelené, modré. Tyto hodnoty jsou pro každou barevnou složku v rozsahu 0 – 255, což odpovídá 8mi bitům. Každý bod se skládá ze tří kanálů, což znamená $3 \cdot 8 = 24$ bitů.

1.5 Vzhled apletu

Aplet je rozdělen na 6 základních částí:

1. Popis
2. Ovládání apletu
3. Vstupní funkce X, Y, Z
4. Zvolení konstant a, b, c
5. přepočtení jednotlivých funkcí aX, bY, cZ

6. Výsledná funkce $aX + bY + cZ$

1.5.1 Popis

Lineární kombinace je pojem, který označuje základní koncept studovaný oborem zvaný lineární algebra. Je to zobecněný pojem pro násobení a sčítání čísel. Tento aplet slouží k názorné ukázce linerní kombinace signálů. Kvůli názornější demonstraci jsou využity obrazy v odstínech šedi. Obraz je v počítačové grafice reprezentován polem bodů. Každý tento bod v rozsahu 0 (černá barva) až 255 (bílá barva) zastupuje hodnotu signálu v daném čase.

V tomto apletu je demonstrován vzorec $ax + by + cz$, kde hodnoty a, b, c jsou konstanty volené uživatelem. Funkce x, y, z reprezentují pole hodnot odstínu (0–255) ze vstupních obrazů X, Y, Z. U lineární kombinace můžou konstanty a, b, c nabývat libovolných hodnot a to kladných i záporných, kdy záporná hodnota se projeví invertováním barev obrazu. Pokud se hodnoty hodnoty konstant a, b, c v součtu rovnají 1 a ani jedna není menší než 0, nazýváme tuto kombinaci konvexní.

Jelikož se u těchto obrazů nemůžeme reálně dostat přes rozsah 0 – 255, jsou výsledné hodnoty přepočteny v příslušném poměru, aby rozsah nepřekročily.

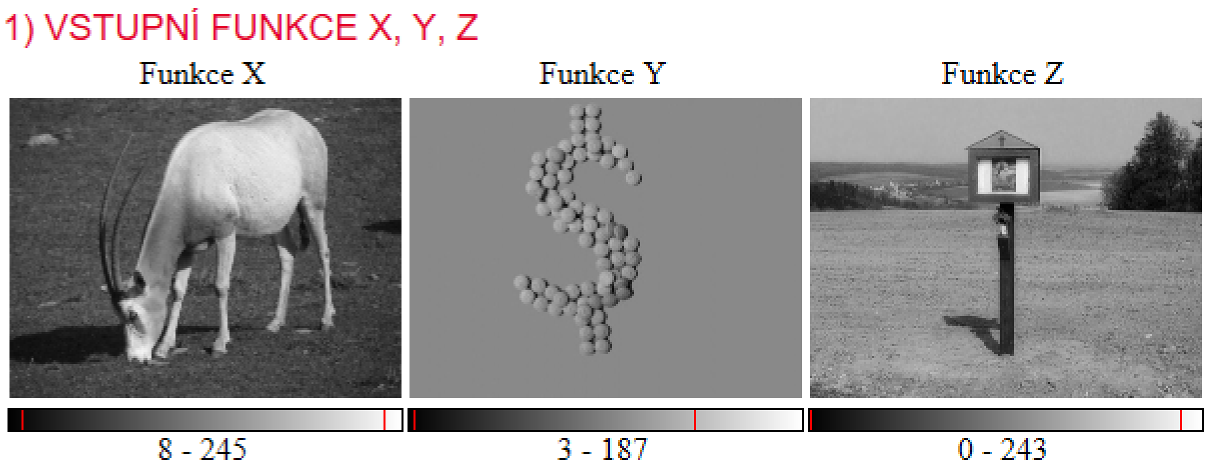
Cílem apletu je vyzkoušet si měnit konstanty a, b, c a pozorovat změny na výsledném zobrazení. Pod obrazy jsou znázorněny i škály stupňů šedi s tím, že maximální i minimální hodnota se zde mění, podle toho jakým číslem obrazy násobíme.

1.5.2 Ovládání apletu

V části 2) Zvolení konstant a, b, c uživatelem se pohybem posuvníku nastaví hodnota příslušné konstanty, při kombinaci lineární v rozsahu -50 až 50. Tento rozsah byl zvolen jako dostatečný pro názornost apletu. U kombinace konvexní 0 až 1 a zbylé proměnné se přepočtou automaticky.

1.5.3 Vstupní funkce X, Y, Z

Na obrázku 1.1 je část se vstupními obrazy apletu. Vstupní obrazy jsou 3. Dole pod každým obrázkem se nachází černobílá škála znázorňující zastoupení maximálního a minimálního jasu v obrazu.



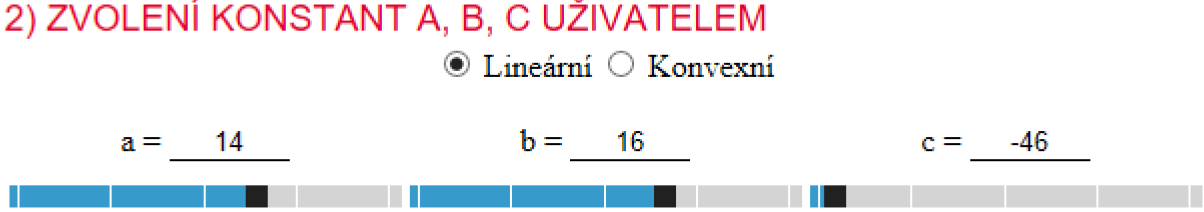
Obr. 1.1: Vstupní obrazy reprezentující funkce X, Y, Z

1.5.4 Zvolení konstant a, b, c

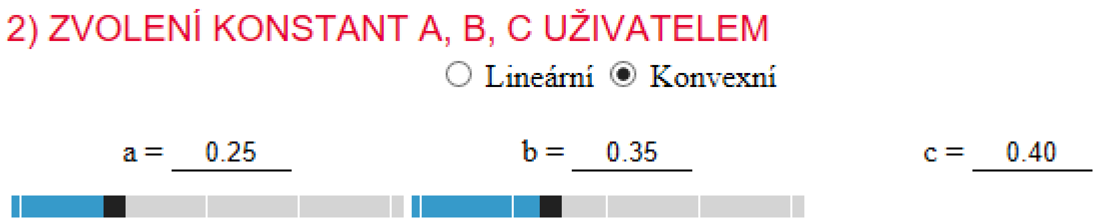
Na obrázcích 1.2 a 1.3 jsou obě varianty nastavení parametrů apletu. Lze vybrat ze dvou možností nastavení konstant a, b, c. První je lineární a to na obrázku 1.2 a druhá konvexní na obrázku 1.3.

Při výběru konvexního nastavení dá součet všech 3 konstant vždy číslo 1. Při změně prvního posuvníku se nastaví konstanta a v rozsahu od 0 v levé krajní poloze do 1 v pravé krajní poloze a přepočítají se konstanty b a c. Při změně druhého posuvníku se nastaví konstanta b v rozsahu od 0 v levé krajní poloze do 1 v pravé krajní poloze a přepočítají se konstanty a a c. Při konvexním nastavení je posuvník pro konstantu c vypnutý, protože není potřebný.

Při výběru lineárního nastavení lze konstanty nastavit jejich posuvníky, a to od hodnoty -50 v levé krajní poloze a $+50$ v pravé krajní poloze. Tento rozsah byl zvolen jako dostatečný pro názornost ukázky apletu.



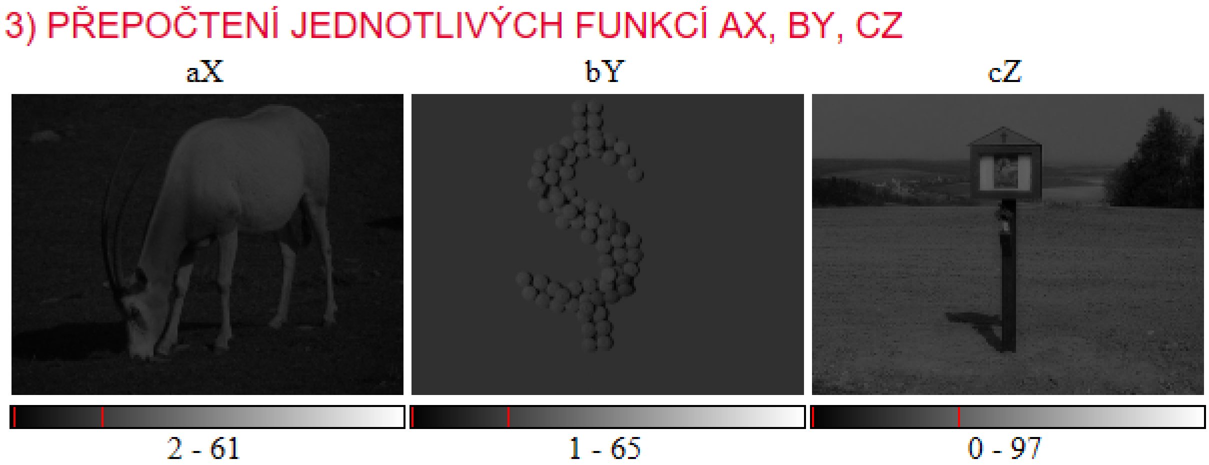
Obr. 1.2: Volení kombinace lineární



Obr. 1.3: Volení kombinace konvexní

1.5.5 Přepočtení jednotlivých funkcí aX, bY, cZ

V této části (obr. 1.4) se přepočítají jednotlivé obrazy podle poměru konstant nastavených výše.



Obr. 1.4: Přepočítané obrazy podle konstant a, b, c

1.5.6 Výsledná funkce aX + bY + cZ

Na obrázku 1.5 je vidět výstupní obraz složený ze všech 3 přepočtených obrazů.

1.6 Programová část

Aplet je naprogramován formou webové stránky s použitím jazyků HTML, CSS a JavaScript. Je rozdělen do 3 souborů:

1. index.html
2. style.css
3. script.js

Obr. 1.5: Výstupní obraz apletu lineární kombinace obrazů

1.6.1 index.html

Tento soubor obsahuje kód vzhledu stránky v jazyce HTML. Byl kladen důraz na jednoduchost stránky pro snadnou orientaci uživatele.

Celá stránka je umístěna ve středu obrazovky a má šířku 600 pixelů. Tato šířka byla vybrána z důvodu nejlepší kompatibility napříč zařízeními.

Ukazatel jasového zastoupení jednotlivých pixelů je realizován dvěma bloky v sobě, což je vidět ve výpisu 1.1 na řádku 3 až 5. Vnější blok má jako pozadí barevný přechod od černé po bílou. Vnitřní blok je průhledný s červenými okraji a nastavuje se pouze jeho levé odsazení a šířka.

Výpis 1.1: Horní boxy v části vstupní obrazy

```
1 <div class="box">
2   <canvas class="image" id="img1"></canvas>
3   <div class="color-bar red-bar">
4     <div class="bar-section" id="img1-red"></div>
5   </div>
6 </div>
```

Všechny obrazy jsou realizovány prvkem <canvas> který je vhodný pro pozdější analýzu a úpravu jeho obsahu.

V části nastavení parametrů jsou dvě přepínací tlačítka, pro výběr lineárního nebo konvexního zadávání proměnných. Při načtení stránky je zaškrtnuto tlačítko lineární, což určuje parametr checked. kód jednoho z přepínačů je vidět ve výpisu 1.2. Na řádku 2 a 3 je napsaná funkce pro aktivaci třetího posuvníku pro proměnnou c, podle vybraného tlačítka.

Výpis 1.2: Tlačítko pro výběr lineárního vstupu

```
1 <input type="radio" id="radio-linear" name="option"
2   value="linear" onchange="if(this.checked)
3   document.getElementById('range-c').disabled = true;"
4   checked>
```

V části nastavení konstant jsou posuvníky, jejichž kód je ve výpisu 1.3. S jejich pomocí se pohodlně nastavují hodnoty proměnných a, b a c. Tato možnost se osvědčila více, než přímé vkládání hodnoty z klávesnice. Při změně hodnoty posuvníku se jeho hodnota ihned projeví v hodnotě proměnné (viz funkce oninput na řádku 4), ale uplatní se až ve chvíli, kdy je tlačítko myši puštěno (viz funkce onchange na řádku 4. Posuvníky mají nastaven rozsah -50 až $+50$ s krokem 1.

Výpis 1.3: Posuvník určující hednotu proměnných a, b, c

```
1 <div class="box">
2   a = <input type="text" value="0" id="value-a" ></input>
3   <input type="range" min="-50" max="50" step="1" id="range-a"
4   oninput="ChangeRange();" onchange="ChangeValue();"></input>
5 </div>
```

1.6.2 style.css

Tento soubor obsahuje kaskádové styly na stránce. Tyto styly určují, jak budou jednotlivé prvky na stránce rozmístěny, jejich vzhled i velikost.

1.6.3 script.js

Výpis 1.4: Funkce načtení a zobrazení obrázku

```
1  this.LoadFile = function (file)
2  {
3    this.img.src = file;
4    this.img.onload = this.imageLoaded.bind(this);
5  }
6
7  this.imageLoaded = function()
8  {
9    this.canvas = document.getElementById(this.element);
10
11   this.canvas.width = 400;
12   this.canvas.height = 300;
13
14   this.ctx = this.canvas.getContext("2d");
15   this.ctx.drawImage(this.img,0,0);
16
17   this.imgData = this.ctx.getImageData(0, 0,
18     this.canvas.width, this.canvas.height);
19   this.Analyze();
20 }
```

Při načtení stránky se nahrají obrázky a zobrazí se v horní části stránky. K tomu slouží funkce LoadFile viz Výpis 1.4. Tato funkce začne nahrávání souboru a ve chvíli kdy je soubor připraven, pošle ho funkci imageLoaded na 7. řádku. Načtený obraz se poté nahraje do komponenty < *canvas* >. Na řádku 17 se projde celý obrázek pixel po pixelu a uloží se do pole imgData v pořadí R, G, B, A, která nabývají hodnot 0–255. Hodnota R reprezentuje červenou složku, G zelenou, B modrou a A průhlednost. Poté se spustí funkce Analyze na řádku 20.

Výpis 1.5: Funkce analýzy obrazu pro zjištění rozsahu barev

```
this.Analyze = function ()
{
  this.rMin = 255; this.rMax = 0;

  for (i = 0; i < this.imgData.data.length; i += 4)
  {
    if (this.imgData.data[i] > this.rMax)
      this.rMax = this.imgData.data[i];
```

```
    if (this.imgData.data[i] < this.rMin)
      this.rMin = this.imgData.data[i];
  }

  document.getElementById(this.element+"-red").
    style.marginLeft = this.rMin/2.55+"%";
  document.getElementById(this.element+"-red").
    style.width = (this.rMax - this.rMin)/2.55+"%";
}
```

Funkce Analyze (viz. výpis 1.5) zjistí zastoupení, respektive nejnižší a nejvyšší hodnotu každé ze 3 složek barev. Na začátku se nastaví nejnižší hodnota na 255 a nejvyšší na 0. Poté se prochází jednotlivé pixely a pokud se najde pixel s nižší hodnotou, uloží se jako minimum a pokud s vyšší, uloží se jako maximum. Na konci funkce se zjištěné hodnoty převedou na procenta a aplikují se na příslušné ukazatele pod jejich obrazem.

Pro složení obrazů je použita funkce Draw. Nejprve se rozlišení obrazu nastaví na hodnotu 400 x 300. Toto rozlišení bylo zvoleno z důvodu velikosti obrazů a době jejich přepočtu a analýze. Není ani malé, aby byly obrazy dostatečně ostré, ani velké aby doba výpočtu nebyla příliš dlouhá.

Po nastavení velikosti se načtou uživatelem nastavené hodnoty, které se aplikují v následujícím kroku při přepočtu obrazových dat. Postupně se prochází jednotlivé pixely a jejich hodnota se nastavuje podle zadaných hodnot.

Když uživatel mění posuvníky u jednotlivých složek, jejich hodnota se přenáší do textových polí u každé z proměnných. Poté se tato hodnota mění podle toho, zda jde o zadávání lineární, nebo konvexní. O tento problém se stará funkce ChangeRange viz výpis 1.6. Při konvexním zadávání musí být součet všech tří proměnných a, b, c číslo 1. Přečte se hodnota z posuvníku a dělí se číslem 50, protože posuvníky mají nastavenou maximální hodnotu právě 50. Při zadávání postupujeme nejprve zadáním hodnoty a, poté hodnoty b, přičemž poslední hodnota c se nám dopočítá sama podle vzorce na řádku 16.

Výpis 1.6: Funkce pro změnu a přepočet proměnných a, b, c

```
function ChangeRange()
{
  var linear = document.getElementById("radio-linear");
  var complex = document.getElementById("radio-complex");

  if(complex.checked)
  {
    a = parseFloat(document.getElementById("range-a").value);
    b = parseFloat(document.getElementById("range-b").value);
    c = parseFloat(document.getElementById("range-c").value);

    a = (a/50).toFixed(2);
    b = (b/50).toFixed(2);
    c = (c/50).toFixed(2);

    c = (1 - a - b).toFixed(2);

    document.getElementById("value-a").value = a;
    document.getElementById("value-b").value = b;
    document.getElementById("value-c").value = c;
  }

  if(linear.checked)
  {
    document.getElementById("value-a").value =
      document.getElementById("range-a").value;
    document.getElementById("value-b").value =
      document.getElementById("range-b").value;
    document.getElementById("value-c").value =
      document.getElementById("range-c").value;
  }
}
```

2 METODA NEJMENŠÍCH ČTVERCŮ A LINEÁRNÍ REGRESE

2.1 Metoda nejmenších čtverců

Je to matematicko-statistická metoda. Slouží k aproximaci řešení soustav, kde je více rovnic, než neznámých (tzv. přeurčených soustav). Pojem "nejmenší čtverce"znamená, že výsledné řešení má minimalizovat součet čtverců odchylek vůči každé rovnici. Používá se pro řešení nekompatibilních soustav lineárních rovnic.

Nejjednodušší aplikace této metody je například prokládání (aproximace) naměřených dat přímkou. Data lze prokládat i parabolou, nebo obecným polynomem libovolného stupně. Tato metoda mé velmi široké využití napříč vědními obory. Jsou to například statistika, ekonomie, geodézie, zpracování signálů, teorie řízení a další.

Obecně tato metoda slouží k eliminaci chyb.

2.2 Lineární regrese

Je to matematická metoda sloužící k proložení souboru bodů v grafu přímkou. U naměřených bodů se předpokládá, že jejich x-ové souřadnice jsou správné, zatímco y-ové se mohou lišit v závislosti na chybě měření. Pokud tyto body proložíme přímkou, tak vznikne mezi body a přímkou určitá odchylka. Podstatou lineární regrese je nalezení takové přímky, aby součet druhých mocnin těchto odchylek byl co nejmenší.

2.3 Aproximace přímkou

Uvažujeme funkční závislost: $f(x) = ax + b$. Součet čtverců pak bude následující

$$S(a,b) = \sum_{i=1}^{n} [f(x_i) - y_i]^2 = \sum_{i=1}^{n} (ax_i + b - y_i)^2 \quad [6] \tag{2.1}$$

kde $[x_i, y_i]$ jsou souřadnice aproximovaných bodů.

Jejím řešením pro konkrétní hodnoty x_i a y_i dostaneme hodnoty parametrů a a b.

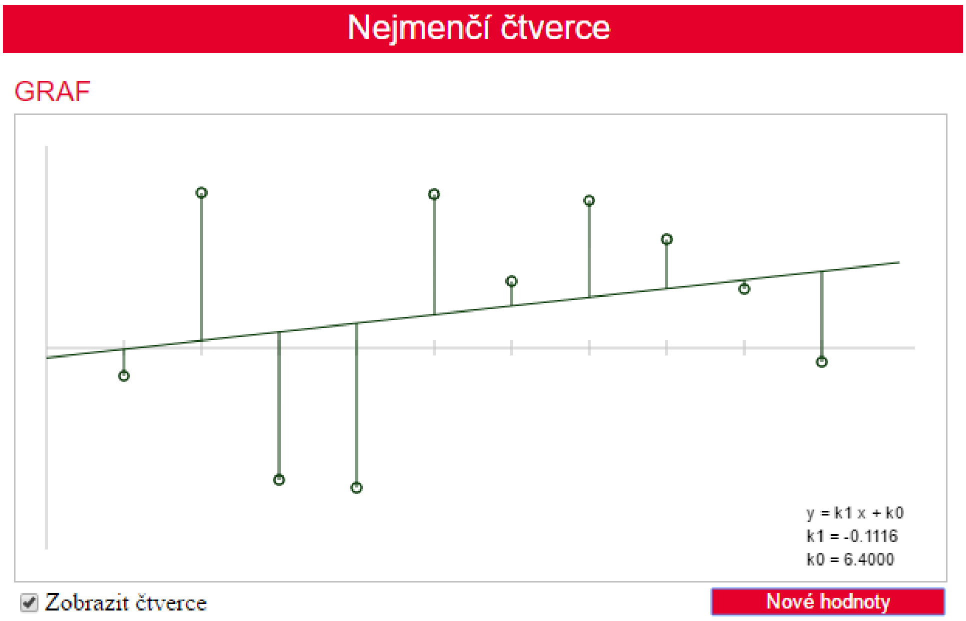
$$a = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - \left(\sum x_i\right)^2} \quad [6] \tag{2.2}$$

$$b = \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{\sum x_i^2 - (\sum x_i)^2} \quad [6] \tag{2.3}$$

Podobný postup lze aplikovat na jakýkoliv druh závislosti i více proměnných.

2.4 Vzhled

Hlavní část vzhledu je graf, obsahující body, které uživatel sám zadá. Pod grafem bude tlačítko, na zobrazení nejmenších čtverců. Po zaškrtnutí políčka vlevo dole se zobrazí jedna strana čtverce mezi bodem a přímkou jak je vidět na obrázku 2.1. Toto zobrazení je kvůli přehlednosti, kdy při větší velikosti čtverců se jednotlivé části překrývaly.



Obr. 2.1: Graf appletu metody nejmenších čtverců

2.5 Funkcionalita

Při kliknutí na tlačítko nové hodnoty se vygenerují náhodné body v grafu a z jejich hodnot se poté vypočte rovnice přímky, která se zobrazí. Vpravo dole je vidět rovnice této přímky i s jejími koefeicienty.

3 DISKRÉTNÍ LINEÁRNÍ KONVOLUCE VE 2D

3.1 Konvoluce

Je to matematický operátor zpracovávající dvě funkce. Spojitá konvoluce dvou funkcí $f(x)$ a $g(x)$ je definována vztahem:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\alpha)g(x-\alpha)d\alpha \quad [7] \tag{3.1}$$

Funkce $g(x)$ se označuje jako konvoluční jádro. Hodnota konvoluce funkce f s jádrem v bodě x je integrál ze součinu funkce f s otočenou funkcí konvolučního jádra posunutou do bodu x.

Pokud jde o konvoluci při zpracování obrazu, je funkce $f(x)$ většinou zkoumaný obrázek a funkce $g(x)$ nějaký filtr.

3.2 Diskrétní konvoluce

Pojem diskrétní konvoluci si vysvětlíme na příkladu. Vezmeme dvě konečné řady, obsahující nějaké hodnoty. Na pozici neexistujících prvků si představíme nuly. Výsledná řada je vždy o jeden prvek kratší než součet délek obou řad.

Konvoluce dvou řad:
(a, b, c, d) $*(e, f, g) =$

$$\begin{array}{llllll}
= (a \cdot e) & (a \cdot f) & (a \cdot g) & & & \\
 & (b \cdot e) & (b \cdot f) & (b \cdot g) & & \\
 & & (c \cdot e) & (c \cdot f) & (c \cdot g) & \\
 & & & (d \cdot e) & (d \cdot f) & (d \cdot g) \\
\hline
\end{array}$$

(součet prvků)
Konkrétní příklad:
(1, 2, -2, -1) $*(1, -1, 2) =$

$$\begin{array}{rrrrrr}
= 1 & -1 & 2 & & & \\
 & 2 & -2 & 4 & & \\
 & & -2 & 2 & -4 & \\
 & & & -1 & 1 & -2 \\
\hline
\end{array}$$

(1, 1, –2, 5, –3, –2)

3.3 Užití v počítačové grafice

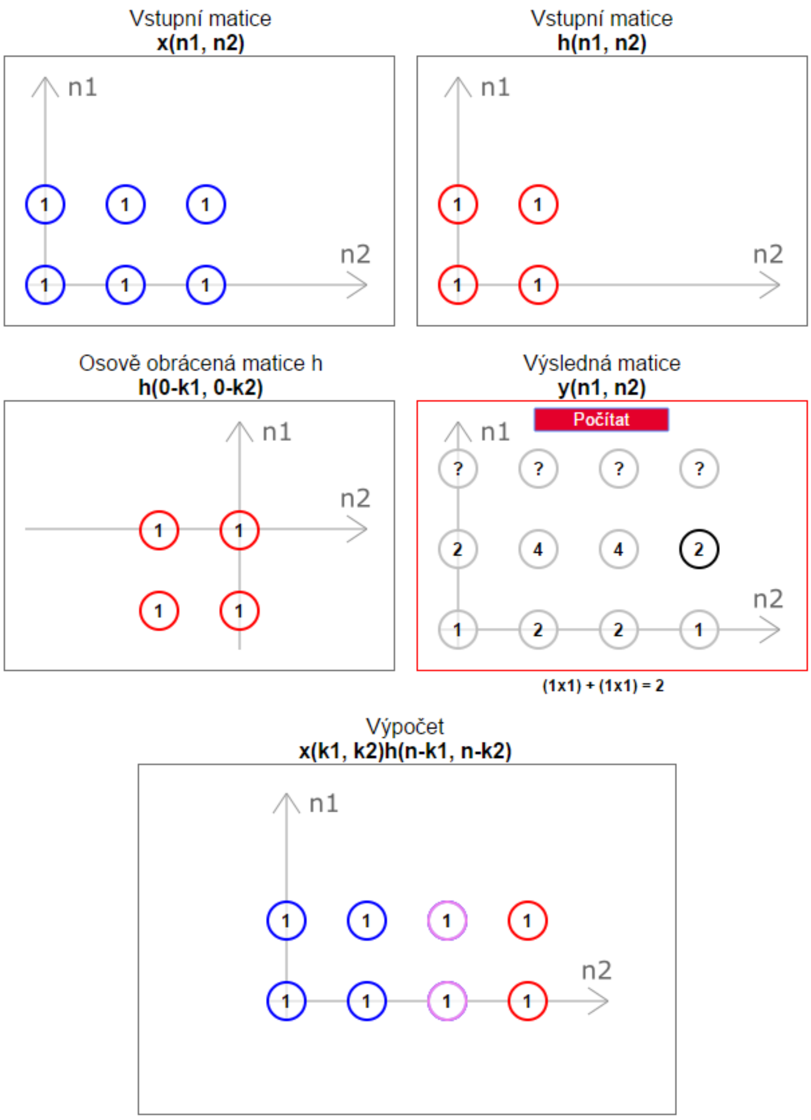
Konvoluce se často používá při algoritmech zpracování dvourozměrného diskrétního obrazu v počítačové grafice. Vzorec diskrétní konvoluce má potom tvar:

$$(f * h)(x, y) = \sum_{i=-k}^{k} \sum_{j=-k}^{k} f(x - i, y - j) \cdot h(i, j) \quad [7] \tag{3.2}$$

V tomto případě chápeme jádro konvoluce jako tabulku (konvoluční masku). Tu položíme na příslušné místo obrazu.

3.4 Vzhled

V horní části se nachází dvě vstupní matice, které uživatel vyplní viz obrázek 3.1. Poté se matice h osově obrátí podle osy n1 a n2. ve výsledné matici se poté postupně zobrazují jednotlivé vypočtené hodnoty a pod ní konkrétní vzorec výpočtu dané hodnoty. Po kliknutí na tlačítko Počítej se začnou po krocích vyplňovat jednotlivá políčka výsledné matice. Ve spodní matici je vidět barevné odlišení jednotlivých hodnot. Políčka která se překrývají se zvýrazní fialově a z těch se dělá samotný výpočet.



Obr. 3.1: Vstupní obrazy appletu lineární konvoluce

4 INTERPOLACE

4.1 Interpolace

Interpolace (lat. inter-polare, vylepšit vkládáním) v numerické matematice znamená nalezení přibližné hodnoty funkce v nějakém intervalu, je-li její hodnota známa jen v některých jiných bodech tohoto intervalu. Používá se v případě, že hodnoty funkce v určitých bodech intervalu jsou buďto uvedeny v tabulce, anebo získány měřením.

4.2 Definice

Mějme funkci $f(x)$, jejíž hodnota je známa v bodech $f(x_0), f(x_1), \dots f(x_n)$. Interpolace znamená nalezení funkční hodnoty $f(x)$, pokud platí, že $x_0 < x < x_n$

4.3 Interpolační křivka

Někdy se interpolací rozumí proložení bodů $f(x_0), f(x_1), \dots f(x_n)$ analytickou křivkou, která pak umožňuje jednoduchý výpočet funkčních hodnot ve všech mezilehlých bodech. Podle počtu známých bodů n se pak nejčastěji používá:

4.4 Lineární interpolace

Nejjednodušší a nejčastěji používaná lineární interpolace (někdy také interpolace lineárním splajnem) spočívá v proložení dvou sousedních bodů přímkou.

Pro $x_0 < x_i < x_1$ platí, že:

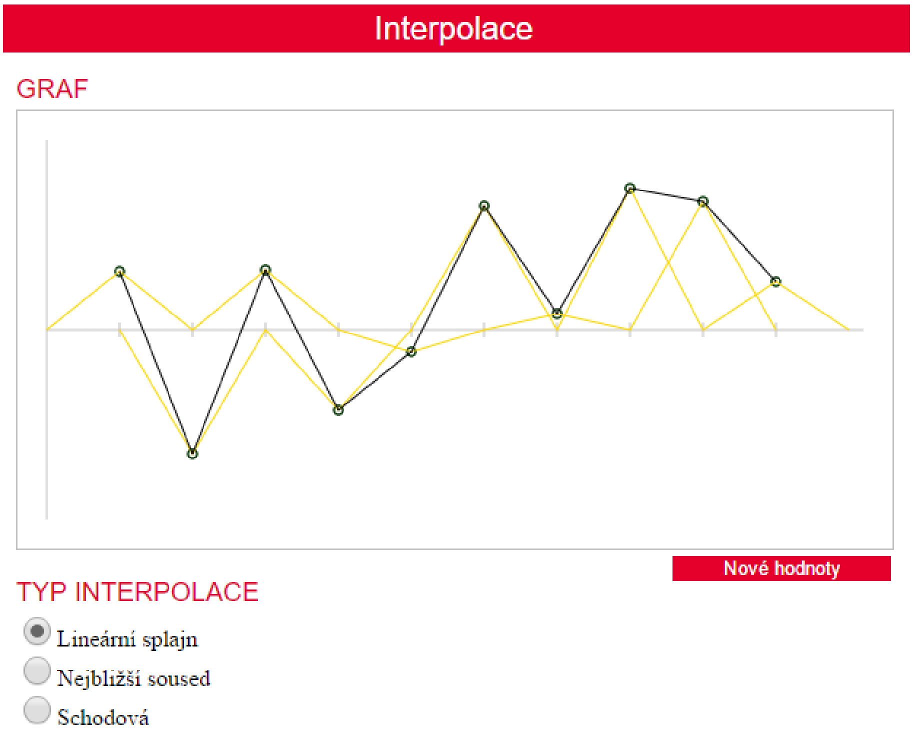
$$f(x) = f_0 + \frac{f_1 - f_0}{x_1 - x_0}(x - x_0) \quad [8] \tag{4.1}$$

- pro $n = 2$ lineární interpolace (přímkou)
- pro $n = 3$ kvadratická interpolace (parabolou nebo kružnicí)
- pro $n > 3$ interpolace polynomem n-tého stupně; pro výpočet koeficientů tohoto polynomu se nejčastěji používá Čebyševova metoda.

4.5 Vzhled

Ve vrchní části okna se nachází graf, ve kterém půjde vybrat kde budou umístěny body pro interpolaci. Pod grafem se nachází výběr příslušné interpolace.

Obr. 4.1: Vzhled uživatelského rozhraní appletu interpolace



4.6 Funkcionalita

Existuje mnoho druhů interpolací. Například lineárním splajnem, Lagrangeova, Newtonova a spoustou dalších funkcí. Zde máme pro demostraci některé z nich. Po kliknutí na tlačítko Nové hodnoty se v grafu vygenerují náhodné body. Ty se následně interpolují zvolenou funkcí.

5 ZÁVĚR

V této práci byly zhotoveny 4 aplety pro názornou demostraci dané problematiky. Jsou to 1/ lineární kombinace obrazů, 2/ metoda nejmenších čtverců a lineární regrese, 3/ diskrétní lineární konvoluce ve 2D, 4/ interpolace v 1D. Při spouštění z lokálního disku funguje aplet lineární kombinace kvůli bezpečnostním opatřením prohlížečů v dané době testování pouze na prohlížeči Edge a Firefox. Při nahrání na server však načítá obrázy bez problému jakýkoliv jiný z testovaných prohlížečů.

LITERATURA

[1] ŽÁRA, Jiří. *Moderní počítačová grafika.* 2., přeprac. a rozš. vyd. Praha: Computer Press, 2004. ISBN 80-251-0454-0.

[2] ANDĚL, J.: *Matematická statistika.* 1. vyd. Praha: SNTL/ALFA, 1978.

[3] PYTLÍČEK, Jiří. *Lineární algebra a geometrie.* Vyd. 3. V Praze: České vysoké učení technické, 2008. ISBN 978-80-01-04063-8.

[4] *Wikipedie: Otevřená encyklopedie: Lineární kombinace* [online]. 2016 [citováno 12. 12. 2016]. Dostupný z WWW: <`https://cs.wikipedia.org/w/index.php?title=Line%C3%A1rn%C3%AD_kombinace&oldid=13791139`>.

[5] Wikipedie: Otevřená encyklopedie: Metoda nejmenších čtverců [online]. 2015 [citováno 13. 12. 2016]. Dostupný z WWW: <`https://cs.wikipedia.org/w/index.php?title=Metoda_nejmen%C5%A1%C3%ADch_%C4%8Dtverc%C5%AF&oldid=13169293`>.

[6] Wikipedie: Otevřená encyklopedie: Lineární regrese [online]. 2016 [citováno 13. 12. 2016]. Dostupný z WWW: <`https://cs.wikipedia.org/w/index.php?title=Line%C3%A1rn%C3%AD_regrese&oldid=13760161`>.

[7] Wikipedie: Otevřená encyklopedie: Konvoluce [online]. 2016 [citováno 13. 12. 2016]. Dostupný z WWW: <`https://cs.wikipedia.org/w/index.php?title=Konvoluce&oldid=14298850`>.

[8] Wikipedie: Otevřená encyklopedie: Interpolace [online]. 2016 [citováno 13. 12. 2016]. Dostupný z WWW: <`https://cs.wikipedia.org/w/index.php?title=Interpolace&oldid=14039235`>.